

REMARKS

The examiner is thanked for the performance of a thorough search. By this amendment, Claims 1, 3, 12, 14, 23, 25-34, and 49-51 have been amended. Claims 2, 13, and 24 are cancelled. Claims 52-65 are added. Hence, Claims 1, 3-12, 14-23, and 25-64 are pending in the application.

New Claim 52 is an apparatus claim that recites the limitations of present Claim 34. New Claim 61 is a computer-readable storage claim that recites the limitations of present Claim 34. New Claims 54-57 are apparatus claims that recite the limitations of present Claims 1, 3-5, respectively. New Claims 58-62 are apparatus claims that recite the limitations of present Claims 34-38, respectively. New Claims 63 and 64 are apparatus claims that recite the limitations of Claims 43 and 45, respectively.

The amendments to the claims as indicated herein do not add any new matter to this application. Furthermore, amendments made to the claims as indicated herein have been made to exclusively improve readability and clarity of the claims and not for the purpose of overcoming alleged prior art.

Each issue raised in the Office Action mailed October 13, 2006 is addressed hereinafter.

I. ISSUES NOT RELATING TO CITED ART

A. 35 U.S.C. § 101 REJECTION

Claims 23-33 and 49-51 stand rejected under 35 U.S.C. § 101 as allegedly directed to non-statutory subject matter. By this amendment, Claims 23-33 and 49-51 have been amended to recite “computer-readable storage medium.” The Office has issued over 5,000 patents that recite at least one claim directed towards a computer-readable storage medium.

For example, issued U.S. Patents 7,065,755, 7,065,740, and 7,065,715 each recite at least one claim directed towards a computer-readable storage medium. Moreover, a computer-readable storage medium is clearly an article of manufacture, which is one of the four statutory categories of patentable subject matter under 35 U.S.C. § 101. The amendments to Claims 23-33 and 49-51 are made to clarify that a computer-readable storage medium stores one or more sequences of instructions that may be executed by one or more processors of a machine. The amendments to Claims 23-33 and 49-51 are not made to disclaim any embodiments where instructions are transmitted or received over a transmission media prior to being stored on the computer-readable storage medium.

B. 35 U.S.C. § 112(2) REJECTION

Claims 4 and 5 stand rejected under 35 U.S.C. 112(2) as allegedly being indefinite for not reciting (in Claim 4) what action is taken if no processes on a node vetoed a software update. Claim 5 depends on Claim 4.

Claim 4 is not required to recite what action is taken if no processes on a node veto a software update. Claim 4 is directed to what action(s) are taken if any processes of a node have vetoed a software update. If Claim 4 was required to recite what occurred if no processes on a node vetoed a software update, then such an amendment would substantially narrow the scope of the claim. Contrary to the interpretation by the Office Action on page 3 that the node “just returns the results of the software update if no veto is taken,” the method recited in Claim 4 covers the situation, for example, where a lack of response by a node after a certain period of time is an indication to the master node that no processes on a node vetoed the software update.

Furthermore, the definiteness requirement of 35 U.S.C. § 112(2) only addresses whether what is recited in the claims is understandable to a skilled artisan. The definiteness requirement is not a basis for requiring an Applicant to recite a particular step.

It is important to note that apparatus and computer-readable medium Claims 15-16 and 26-27, which correspond to Claims 4-5, were not rejected.

Therefore, removal of the 35 U.S.C. 112(2) rejection with respect to Claims 4 and 5 is respectfully requested.

## II. ISSUES RELATING TO CITED ART

Claims 1-10, 12-21, 23-32, 34-41, and 43-51 stand rejected under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 7,013,461 issued to Hellerstein et al. ("*Hellerstein*"). This rejection is respectfully traversed.

Claims 11, 22, 33, and 42 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over *Hellerstein*. This rejection is respectfully traversed.

### A. CLAIM 1

Present Claim 1 recites:

A method of software change modeling for nodes in a distributed network of nodes, the method comprising the computer-implemented steps of:  
providing a master node;  
receiving a software update for a node on said master node;  
wherein the software update contains a software package or a set of software packages;  
wherein a software package contains at least one software module with corresponding software dependency information;  
wherein said master node notifies said node that a software update is being requested;  
wherein **said master node passes said node** identities of software package(s) to be updated and **software dependency information**; and

wherein **said node determines, using the software dependency information, running processes on said node that will be affected by the software update.** (emphasis added)

At least the above-bolded features of present Claim 1 are not taught or suggested by *Hellerstein*.

1. *Overview of Hellerstein*

In *Hellerstein*, a service distribution server (SDS) 205 prepares a software package for distribution based on (1) policy data, (2) dependency information indicating requisites for a service provided by the software package, and (3) configuration information for each region. The SDS 205 distributes the software package to one or more region servers 203, each representing a region of a distributed network. The software package is customized at each region server 203 based on (1) regional distribution policies, (2) dependency information specific to one or more roles performed by one or more target machines 202 in that region, and (3) individual target machine configuration information. Each region server 203 then distributes a customized software package to one or more target machines 202 in the respective region (see Abstract and FIG. 2).

2. *Hellerstein fails to teach or suggest passing software dependency information to a node*

Present Claim 1 is directed to a fundamentally different approach than *Hellerstein*. Claim 1 recites that a master node “passes said node identities of software package(s) to be updated **and software dependency information**” (emphasis added). The Office Action equates (1) the “master node” of Claim 1 with the SDS 205 of *Hellerstein* and (2) the “node” of Claim 1 with a target machine 202 of *Hellerstein*. In order to teach or suggest the above quoted feature of Claim 1, the SDS 205 must pass a target machine 202 software dependency information.

However, **neither the SDS 205 nor a region server 203 passes a target machine 202 dependency information.** *Hellerstein* teaches that different dependency information is used by SDS 205 and region servers 203 to determine how to prepare and/or customize a software package. In contrast, the master node of Claim 1 passes the node of Claim 1 software dependency information, in addition to the identities of software packages to be updated.

3. *Hellerstein fails to teach or suggest that the node uses software dependency information to determine processes on the node that will be affected by the software update*

Claim 1 also recites that the **node** (i.e., that receives the identities of software packages to be update and software dependency information) **uses the recited software dependency information to determine running processes on the node that will be affected by the software update.** *Hellerstein* fails to teach or suggest this feature of Claim 1. As discussed previously, the target machine 202 **does not** receive software dependency information. Therefore, the target machine 202 **could not use software dependency information** to determine running processes on the target machine 202 that will be affected by the software update.

The Office Action cites col. 8, lines 54-58 of *Hellerstein* for disclosing that “said node determines running processes...on said node that will be affected by the software update using the software dependency.” That portion of *Hellerstein* states: “For each end point machine within a domain, the RS [region server 203] performs an inventory scan to determine the environment of the machine. If the result satisfies the dependency requirements of the SP [new software package], the end machine is a potential target.” This statement is entirely consistent with the Abstract of *Hellerstein* and the rest of the description of *Hellerstein*. Even if the recited “running processes” of Claim 1 could be equated with the “environment of the machine” (as the

Office Action asserts), *Hellerstein* fails to teach or suggest that the **target machine** (i.e., the alleged node) **determines** the environment of the machine, much less that the target machine uses software dependency information to make such a determination. Furthermore, *Hellerstein* fails to mention or make reference to running processes on a target machine, much less running processes “that will be affected by the software update.”

Based on the foregoing, *Hellerstein* fails to teach or suggest numerous features of Claim 1 discussed above. It is therefore respectfully submitted that Claim 1 is patentable over *Hellerstein*. Removal of the 35 U.S.C. § 102(b) rejection with respect to Claim 1 is respectfully requested.

B. CLAIMS 12, 23, AND 54

Independent Claims 12, 23, and 54 are either an apparatus claim or a computer-readable storage medium claim. Each of Claims 12, 23, and 54 recite features discussed above that distinguish Claim 1 from *Hellerstein*. Therefore, each of Claims 12, 23, 54 is allowable for the reasons given above with respect to Claim 1.

C. CLAIM 34

Present Claim 34 recites:

A method of software change modeling of networked nodes on a computer system, the method comprising the computer-implemented steps of:  
providing a **software update simulator** on said computer system;  
**simulating processes from at least one node on said computer system;**  
wherein **each functional process that is simulated is a minimal version of a functional process that runs on said node;** and  
receiving a software update for said node by said software update simulator;  
wherein the software update contains a software package or a set of software packages;  
wherein a software package contains at least one software module with corresponding software dependency information;

wherein said software update simulator notifies a control process for said node that a software update is being requested; and  
wherein said software update simulator passes said control process identities of software package(s) to be updated **and software dependency information**. (emphasis added)

At least the above-bolded features of present Claim 34 are not taught or suggested by *Hellerstein*.

1. *Hellerstein fails to teach or suggest the software update simulator of Claim 34*

The Office Action asserts that the region server 203 of *Hellerstein* “simulates the functions of the targets 202 when viewed from the service distribution server [205]” (page 15). In order for this statement to be true, region server 203 would have to perform (at least similar) functions of a target machine 202. **This is simply untrue.** The SDS 205 does not treat a region server 203 as a target machine 202. The SDS 205 may not have to know how region server 203 performs its functions, but SDS 205 “knows” that region server 203 is not a target machine 202. Consequently, the SDS 205 does not act as if region server 203 is a target machine 202.

It is respectfully submitted that the region server 203 **does not simulate any processes** from any target machine 202 (i.e., the alleged node). Not only is there no mention of processes from a target machine 202, the Office Action **does not provide support** for the allegation that region server 202 simulates functions (i.e., the alleged processes) of target machine 202.

Paragraph 186 of the present application states: “Software change modeling allows a user **to simulate a software change** to a system. The user can discover what effects a software update will have on a node or a set of nodes **without actually impacting the router or computer network.**” In contrast, the invention of *Hellerstein* discover the effects of a software

package **after the software package is distributed** to the appropriate target machines 202.

Thus, *Hellerstein* does not teach or suggest any form of simulation.

2. *Hellerstein fails to teach or suggest “minimal version of a functional process”*

The Office Action also asserts that the “legacy version” mentioned at col. 5, line 12 of *Hellerstein* discloses “wherein each functional process from said node is a minimal version of a functional process that runs on said node.” The applicable portion of *Hellerstein* merely states: “‘Updateable’ flag: this allows a user to prevent the updating of an installed package, e.g., when a legacy version of a word processor must be used and must not be updated” (col. 5, lines 11-13). This portion of *Hellerstein* fails to teach or suggest this feature of Claim 34 for at least two reasons. First, the term “**legacy version**” refers to an older version of a software package where a newer version is available. On the other hand, a **minimal** version of a functional process is orthogonal to the newness or oldness of the functional process. Second, in order to read on this feature of Claim 34, **the legacy version** (e.g., of a word processor) **would at least have to be simulated** from a node in a network of nodes in a computer system – a logical implication that is inconsistent with the teachings of both *Hellerstein* and Claim 34.

3. *Hellerstein fails to teach or suggest “said software update simulator passes said control process ...software dependency information”*

Similarly with Claim 1, *Hellerstein* fails to teach or suggest that the SDS 205 or a region server 203 (i.e., the alleged software update simulator) passes software dependency information to a target machine 202 (i.e., the alleged node). The SDS 205 and a region server 203 use different dependency information to prepare and customize, respectively, a software package for distribution to a target machine 202. Thus, not only do SDS 205 and a region server 203 fail to



pass software dependency information to a target machine 202, there would be no reason to do so.

Based on the foregoing, *Hellerstein* fails to teach or suggest numerous features of Claim 34 discussed above. It is therefore respectfully submitted that Claim 34 is patentable over *Hellerstein*. Removal of the 35 U.S.C. § 102(e) rejection with respect to Claim 34 is respectfully requested.

D. CLAIMS 52, 53, AND 58

Independent Claims 52, 53, and 58 are either an apparatus claim or a computer-readable storage medium claim. Each of Claims 52, 53, and 58 recite features discussed above that distinguish Claim 34 from *Hellerstein*. Therefore, each of Claims 52, 53, and 58 is allowable for the reasons given above with respect to Claim 34.

E. CLAIM 43

Present Claim 43 recites:

A method of software change modeling of nodes in a network of nodes on a computer system, the method comprising the computer-implemented steps of:

providing a **software update simulator** on said computer system;

wherein said software simulator runs software components normally run on a master node in the network of nodes;

**wherein a user loads a node's current software configuration into said software simulator by loading current software modules installed on a node;**

**wherein the user requests a simulation of a software update by loading an updated software image into said simulator;**

wherein the software image contains a set of one or more software packages;

wherein each software package contains at least one software module and software dependency information of said each software package;

**wherein said software simulator calculates the software update's impact on said node using the current software configuration of said node; and**

displaying the calculation's results to the user. (emphasis added)

At least the above-bolded features of present Claim 1 are not taught or suggested by *Hellerstein*. As stated above with respect to Claim 34, a region server 203 cannot be equated with the "software update simulator" of Claim 43.

1. *Hellerstein fails to teach or suggest "wherein a user loads a node's current software configuration into said software simulator by loading current software modules installed on a node"*

The Office Action cites col. 2, lines 64-66 of *Hellerstein* as disclosing "wherein a user loads a node's current software configuration into said software simulator by loading current software modules installed on a node." That portion of *Hellerstein* merely states: "[T]he invention provides a methodology by which an individual user in an enterprise can force installation of a software package by installing the package locally and bypassing the automatic system." At most this portion teaches that a user can install a software package directly on a target machine 202 without going through the SDS 205 or a region server 203. In order to read on this feature of Claim 43, *Hellerstein* would have to teach that a user loads a target machine's 202 (i.e., the alleged node) current software configuration into a region server 203 (i.e., the alleged software simulator). If the "user" of *Hellerstein* loads anything, then at most it is a new software package, **not the current software configuration of a target machine 202.**

Furthermore, at most, the cited portion of *Hellerstein* discloses that a user can load a new software package **into a target machine 202**, whereas Claim 43 would require that the software package is loaded **into a region server 203** (i.e., software simulator).

2. *Hellerstein fails to teach or suggest “wherein the user requests a simulation of a software update by loading an updated software image into said simulator”*

The Office Action asserts that col. 1, line 57 of *Hellerstein* as disclosing “wherein the user requests a simulation of a software update by loading an updated software image into said simulator.” That portion of *Hellerstein* merely states: “[T]here is a request for distribution of the software package from the administrator 10 to a distribution computer system or server 18” in a “conventional software distribution system” (col. 1, lines 44-45 and 57-59). This cited portion merely teaches that a user requests **actual distribution of a software package**. However, Claim 43 recites that the user requests a **simulation of a software update**. *Hellerstein* fails to teach or suggest any remotely equivalent to a simulation of a software update.

3. *Hellerstein fails to teach or suggest “wherein said software simulator calculates the software update’s impact on said node using the current software configuration of said node”*

The Office Action cites col. 9, lines 54-55 of *Hellerstein* as disclosing “wherein said software simulator calculates the software update’s impact on said node using the current software configuration of said node.” That cited portion merely states: “If, after obtaining a new policy definition (step 801), a conflict is detected with existing policies (step 802), [then] the new policy is rejected (step 803)” (col. 9, lines 54-56). The Office Action equates “calculates” of Claim 43 with “detects” in the cited portion of *Hellerstein*. This is incorrect.

“This process [of FIG. 8] deals with the population of the policy repository (209 in FIG. 2)” (col. 9, lines 48-49). The conflict that is detected is between **policies**. Detecting conflict between policies is **completely orthogonal** to calculating an impact of a software update. *Hellerstein* fails to teach or suggest anywhere that an **impact of a software update** is

**calculated**. Furthermore, even if detecting conflicts between policies were similar to calculating the impact of a software update, the region server 203 (i.e., the alleged software update simulator) must detect such policy conflicts. However, because a region server 203 is **not** communicatively coupled to policy repository 209 (see FIG. 2), a **region server 203 could not detect a conflict in policies**, much less an impact of a software update on a node.

Furthermore, even if a region server 203 did calculate such an impact, no where does *Hellerstein* teach or suggest that **the region server 203 uses current software configuration of the node to perform the calculation**.

Based on the foregoing, *Hellerstein* fails to teach or suggest numerous features of Claim 43 discussed above. It is therefore respectfully submitted that Claim 43 is patentable over *Hellerstein*. Removal of the 35 U.S.C. § 102(e) rejection with respect to Claim 43 is respectfully requested.

F. CLAIMS 46, 49, AND 63

Independent Claims 46, 49, and 63 are either an apparatus claim or a computer-readable storage medium claim. Each of Claims 46, 49, and 63 recite features discussed above that distinguish Claim 43 from *Hellerstein*. Therefore, each of Claims 46, 49, and 63 is allowable for the reasons given above with respect to Claim 43.

G. DEPENDENT CLAIMS

The dependent claims (including Claims 11, 22, 33, and 42) not discussed thus far are dependent claims, each of which depends (directly or indirectly) on one of the independent claims discussed above. Each of the dependent claims is therefore allowable for the reasons given above for the claim on which it depends. In addition, each of the dependent claims introduces one or more additional limitations that independently render it patentable. However, due to the fundamental differences already identified, to expedite the positive resolution of this

case, a separate discussion of those limitations is not included at this time. The Applicant reserves the right to further point out the differences between the cited art and the novel features recited in the dependent claims.

### III. CONCLUSIONS & MISCELLANEOUS

For the reasons set forth above, all of the pending claims are now in condition for allowance. The Examiner is respectfully requested to contact the undersigned by telephone relating to any issue that would advance examination of the present application.

A petition for extension of time, to the extent necessary to make this reply timely filed, is hereby made. If applicable, a law firm check for the petition for extension of time fee is enclosed herewith. If any applicable fee is missing or insufficient, throughout the pendency of this application, the Commissioner is hereby authorized to any applicable fees and to credit any overpayments to our Deposit Account No. 50-1302.

Respectfully submitted,

HICKMAN PALERMO TRUONG & BECKER LLP

Dated: January 16, 2007



Daniel D. Ledesma

Reg. No. 57,181

2055 Gateway Place Suite 550  
San Jose, California 95110-1093  
Telephone No.: (408) 414-1080  
Facsimile No.: (408) 414-1076